

Engineering of Data Acquiring Mobile Software and Sustainable End-User Applications

Benton T. Smith
Kennedy Space Center
Major: Computer Engineering
KSC FO, Summer Session
Date: 5th August 2013

Engineering of Data Acquiring Mobile Software and Sustainable End-User Applications

Benton T. Smith¹

*California State University, San Bernardino
San Bernardino, California, 92407*

Nomenclature

<i>AGSM</i>	=	Advanced Ground Systems Maintenance
<i>AIDM</i>	=	AGSM Intelligent Devices Mobile
<i>API</i>	=	Application Programming Interface
<i>IDE</i>	=	Integrated Development Environment
<i>KSC</i>	=	John F. Kennedy Space Center
<i>OS</i>	=	Operating System

Abstract

The criteria for which data acquiring software and its supporting infrastructure should be designed should take the following two points into account: the reusability and organization of stored online and remote data and content, and an assessment on whether abandoning a platform optimized design in favor for a multi-platform solution significantly reduces the performance of an end-user application. Furthermore, in-house applications that control or process instrument acquired data for end-users should be designed with a communication and control interface such that the application's modules can be reused as plug-in modular components in greater software systems. The application of the above mentioned is applied using two loosely related projects: a mobile application, and a website containing live and simulated data. For the intelligent devices mobile application AIDM, the end-user interface have a platform and data type optimized design, while the database and back-end applications store this information in an organized manner and manage access to that data to only to authorized user end application(s). Finally, the content for the website was derived from a database such that the content can be included and uniform to all applications accessing the content. With these projects being ongoing, I have concluded from my research that the applicable methods presented are feasible for both projects, and that a multi-platform design for the mobile application only marginally drop the performance of the mobile application.

¹ KSC NASA Intern; AGSM; John F. Kennedy Space Center; California State University, San Bernardino

I. Introduction

The criteria for which data acquiring software and its supporting infrastructure should be designed should take the following two points into account: the reusability and organization of stored online and remote data and content, and an assessment on whether abandoning a platform optimized design in favor for a multi-platform solution significantly reduces the performance of an end-user application. Furthermore, in-house applications that control or process instrument acquired data for end-users should be designed with a communication and control interface such that the application's modules can be reused as plug-in modular components in greater software systems. The application of the above mentioned is applied using two loosely related projects: a mobile application and a website containing live and simulated data.

II. Projects

Before getting into the details of the two criterion for which software and informational resources should be designed, it is first necessary to understand the projects that are going to be used throughout this document as case studies. The first of such project is a mobile application (AIDM), which receives data from data acquiring systems and allows the user to use various tools to analyze the data. This application emphasizes an architecture of which it can be incorporated into a larger system that in turns passes various signals to a variety of sub-systems.

The second case study is a website for Advanced Ground Systems Maintenance (AGSM). The purpose of this website is that it is to be an informative, interactive, intuitive, and maintainable. In other words, while showcasing AGSM's capabilities and eventually educating the public, the site is also required to be easily maintainable by AGSM as well as Kennedy Space Center's IT web team. For this case study, only one criterion would apply.

III. Criteria A: Maintainability and Organization of Software and Online Resources

The first main criteria is that software and online resources should designed with their use in mind, or in the case of both AIDM and AGSM's web site, with a long term and robust use in mind. For instance, AGSM's website needs to educate and do so in an organized manner.

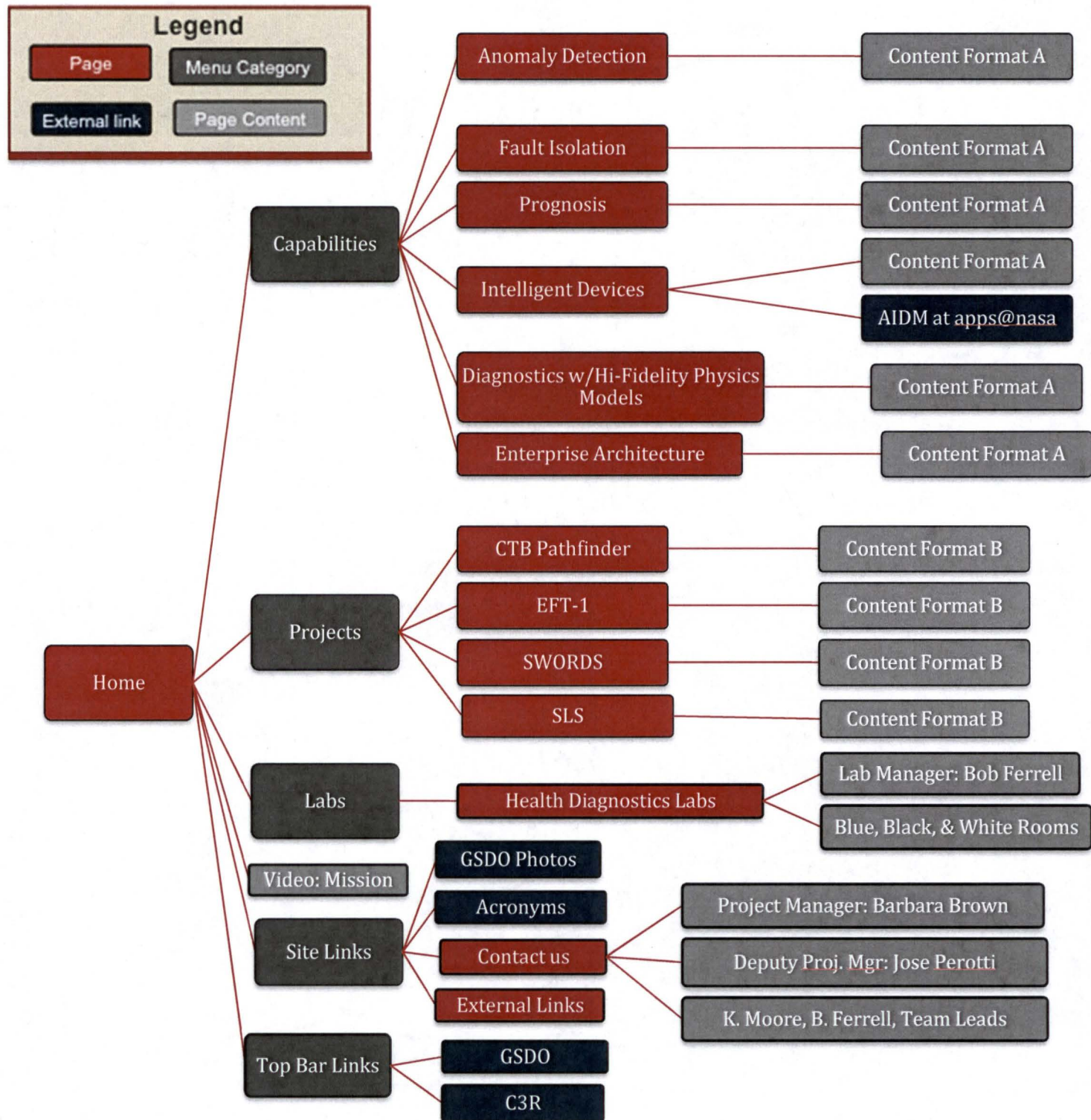


Figure 1A: AGSM Site Map

The website hierarchy of pages and links. Content formats A and B are described in Figure 1B



Figure 1B: Format for Content on an Informational Page

Figure 1A shows the site map for the AGSM website. Notice that all of the pages under the category “Capabilities” follow the same basic structure as outlined by “Content Format A” of Figure 1B, and that those under “Projects” follow “Content Format B”. Creating pages of similar types with a basic formats evens out the learning curve for navigating through and finding information on the site, very much how a dictionary has the words under letter “A” and then formats each of the definitions so that the word comes first and the actual definition (or example) is last: the consistency is subconsciously picked up by the user and then that knowledge is later used when viewing similar pages.

Now for taking reusability into account, AIDM will be used as the case study. This mobile application’s primary function is to display sensory data that is securely broadcasted over a network so that the user can view and analyze the data from an iOS compatible device. This will be achieved by using ICE communication interface to list the device as a subscriber to the sensor data stream, and an intermediate server (Figure 2). The intermediate server will serve as both a lookup service for locating the sensors and for the formatting of the sensory information. Reusability comes into account here: since ICE is a communications library developed for a variety of platforms and programming languages, it allows various systems to connect over a common bus, thereby allowing these sub-systems to communicate with each other without the need for translating and formatting every message from the host sub-system’s language to the language of each subscriber. This will assist with the refurbishment of the communications code as well as that of the mobile application itself, and should easily allow AIDM’s code to be modified to read other types of data broadcasted on the bus by changing the feed that it subscribes to.

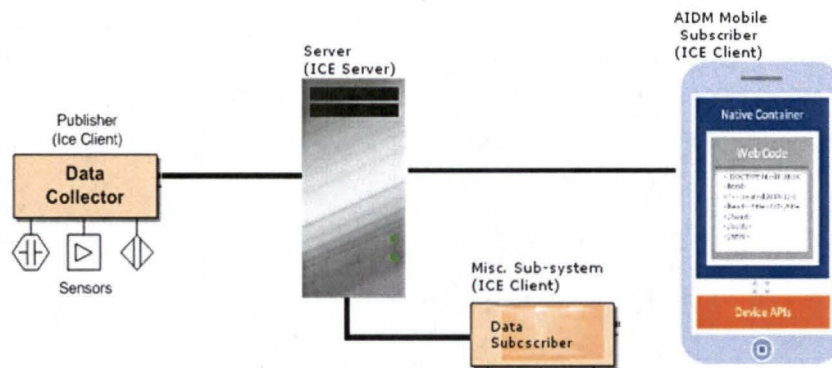


Figure 2: Deployment Diagram for AIDM

IV. Criteria B: Platform Specific vs. Hybrid Design

It is commonly known among software engineers that developing the same software to behave the same on multiple platforms often end up being a hassle; it requires additional time per platform, and additional planning to develop the software for various platforms. In mobile application development, this is quite troublesome since there are various mobile platforms: unlike desktop environments where you can use virtual machines to run applications on various operating systems, mobile devices generally contain neither the computational power nor the software to do so. This problem, however, has a solution in which a few languages are needed in order to develop applications that are uniform across platforms, called hybrid applications. Hybrid applications work by using native, OS specific libraries in order to translate code from the common language to the native language for that particular mobile OS. This has the advantage of eliminating the need to write separate code for various platforms, however it also can affect performance. So the important question is whether to take a hybrid approach or a native (platform specific) approach to engineering software such as AIDM.

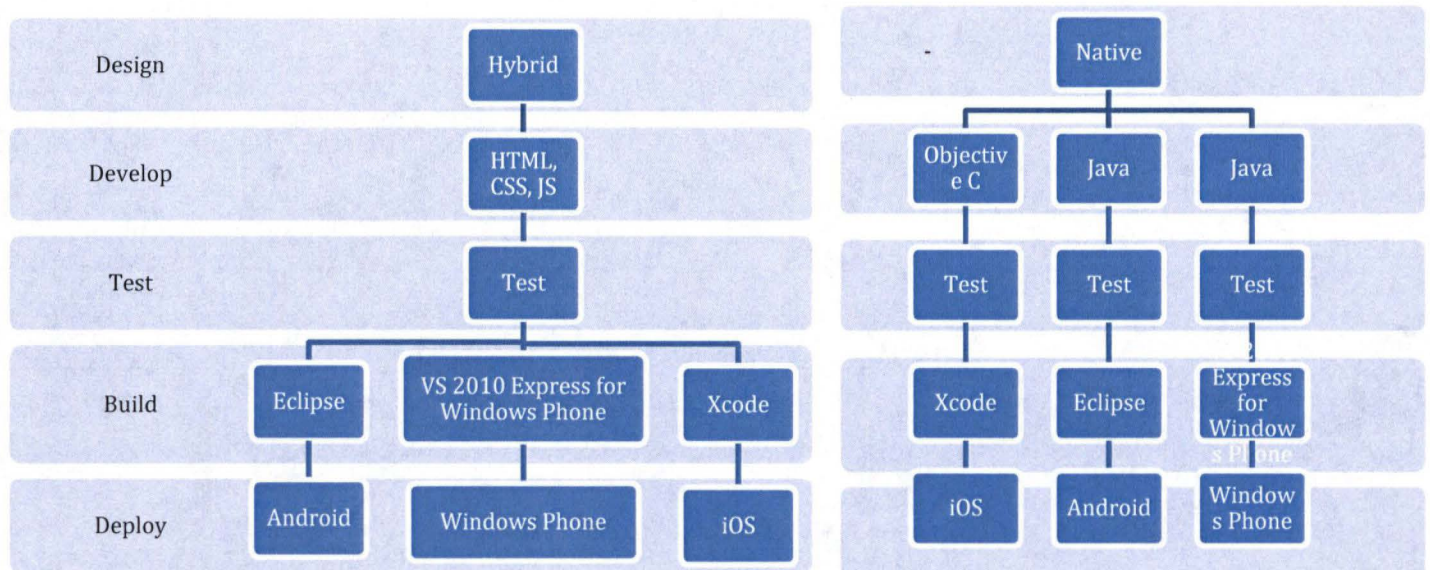


Figure 3: Development cycles of Native and Hybrid Applications

In order to make a valid assessment on which approach to use, the benefits and costs of each approach must be looked at. For example, native applications take longer to develop for multiple platforms due to the necessity of writing code for each platform. Hybrid applications, on the other hand, require little to no native coding and generally require only one or two languages to write code for all platforms¹. The caveat with hybrid applications is that it is not efficient as a native application. Another disadvantage is that hybrid applications require a framework, which varies in functionality and supported platforms. Some hardware, for example, may not be supported, and no hybrid framework supports all APIs and hardware of all the devices and OSes that they cover.

In the case of AIDM, a hybrid approach would be preferred. The overall performance of hybrid over native in this case would be negligible. Since this application is not meant to be graphics intensive, there is very little loss of graphical ability: in fact, the loss is visually nonexistent. Additionally, a good IDE will only include the headers of a framework required for the specific application, thereby reducing the application's file size. Furthermore, most hybrid frameworks, such as Phone Gap and Dojo Mobile, support the APIs and hardware required by AIDM. These components include Wi-Fi, network, and the system sleep routine. Finally, the update frequency of the information presented to the

phone will be reduced in order to prevent the application from depleting the user's data plan; this update frequency limit will also allow enough data to use for generating smooth graphs and charts without overloading memory or requiring too much of the processor's time.

V. Conclusion

AIDM and the AGSM website concepts were both reviewed in order to find the most effective manner for each of their implementation. With AIDM, a cross-platform hybrid approach was taken after an assessment of the performance cost predetermined that a hybrid approach would not significantly affect the overall performance of the mobile application. Furthermore, its design is optimized for reuse with other systems using the ICE communications bus, therefore encompassing the various systems under AGSM. AGSM website is an informative website with a fairly uniform structure throughout the pages of the website. While it encompasses a cornucopia of information, the information is organized and will be easily maintainable. Furthermore, the content on the website can potentially be ported into a mobile application or a mobile website in the future without the need to hardcode the information again.

Acknowledgments

The author gratefully acknowledges Barbara Brown for being the client that requested the web page and provided the specifications, as well as Barbara Brown and Bob Ferrell for equipment for mobile application development on the iOS platform.

The author also gratefully acknowledges Felix A. Soto-Toro for mentoring, as well as Kathryn “Kathy” Moore for providing assistance with resource access and administrative support.

And finally, the author thanks the National Science Foundation (NSF) and KSC Ground Systems Development Operations (GSDO) for funding.

Resources

- ¹Christ, A. M. (2011, October). Bridging the Mobile App Gap. (P. H. Gilbert Miller, Ed.) *Sigma*, 11(1), 27 - 32. Retrieved August 8, 2013, from <http://www.danielna.com-NoblisSigma-DigitalEcosystem.pdf#page=29>